Back End Web Development

**BED Assignment 2 API documentation**

Doo Yan Xu, Ng Wei Jun

horizontal line

# Introduction

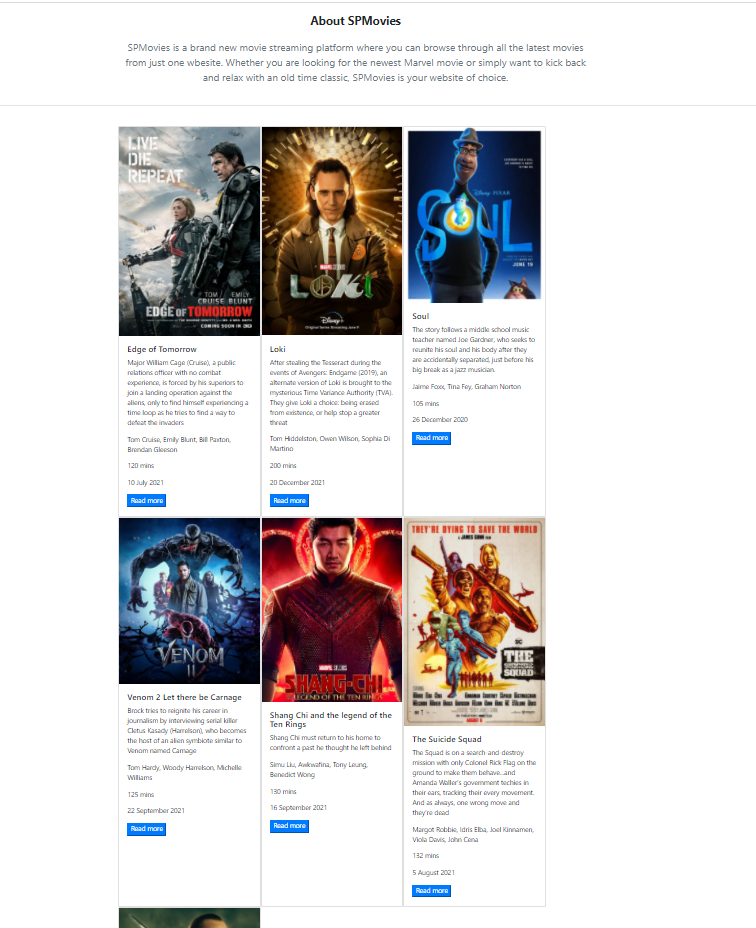
For the assignment, we have created an SPMovies website that consists of both a front-end and back-end server that allows data to be dynamically loaded from a mysql database.

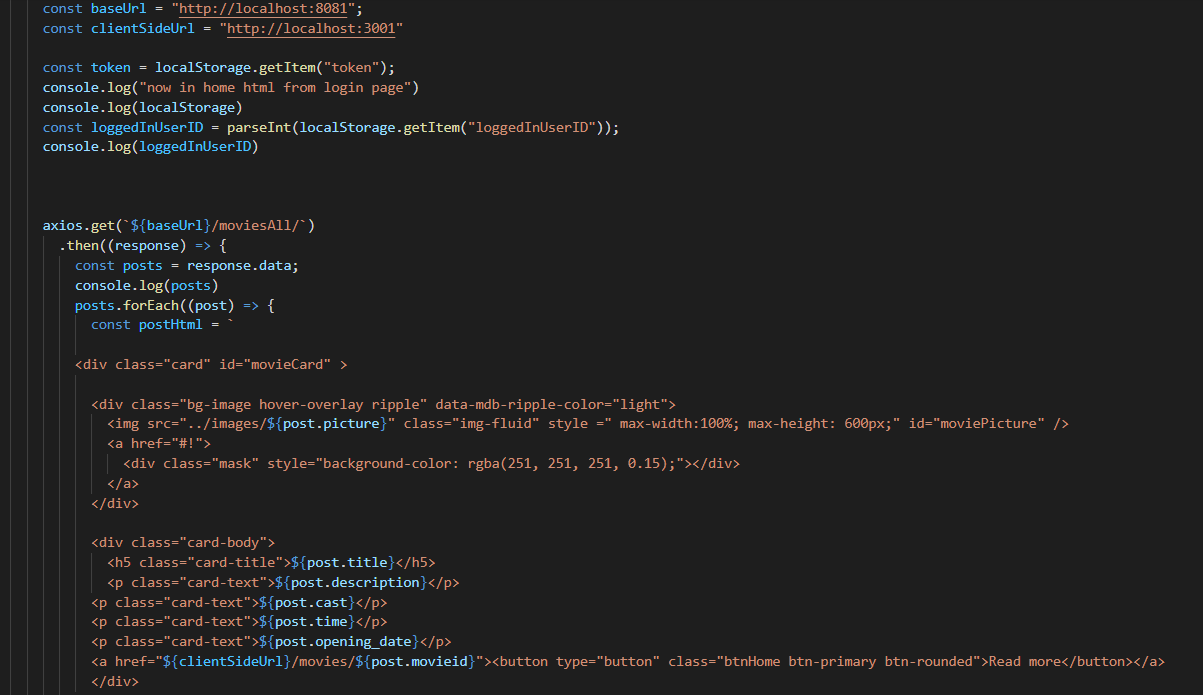
In the website, we have created multiple API’s.

## Get all movies

The home page consists of an API that supports the get all movies function. This will be loaded every time the user visits the home page. This displays all the movies in the database.

Front end for home page

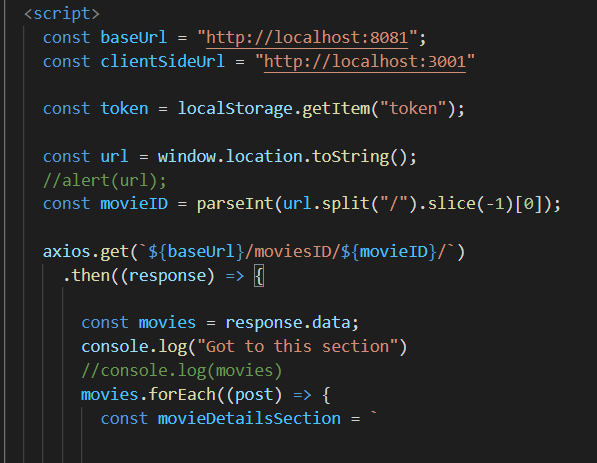




The front end will call the back end and get all movies functions using axios. The retrieved movies will then be displayed on the front end.

## Get movie by ID/ Get movie reviews

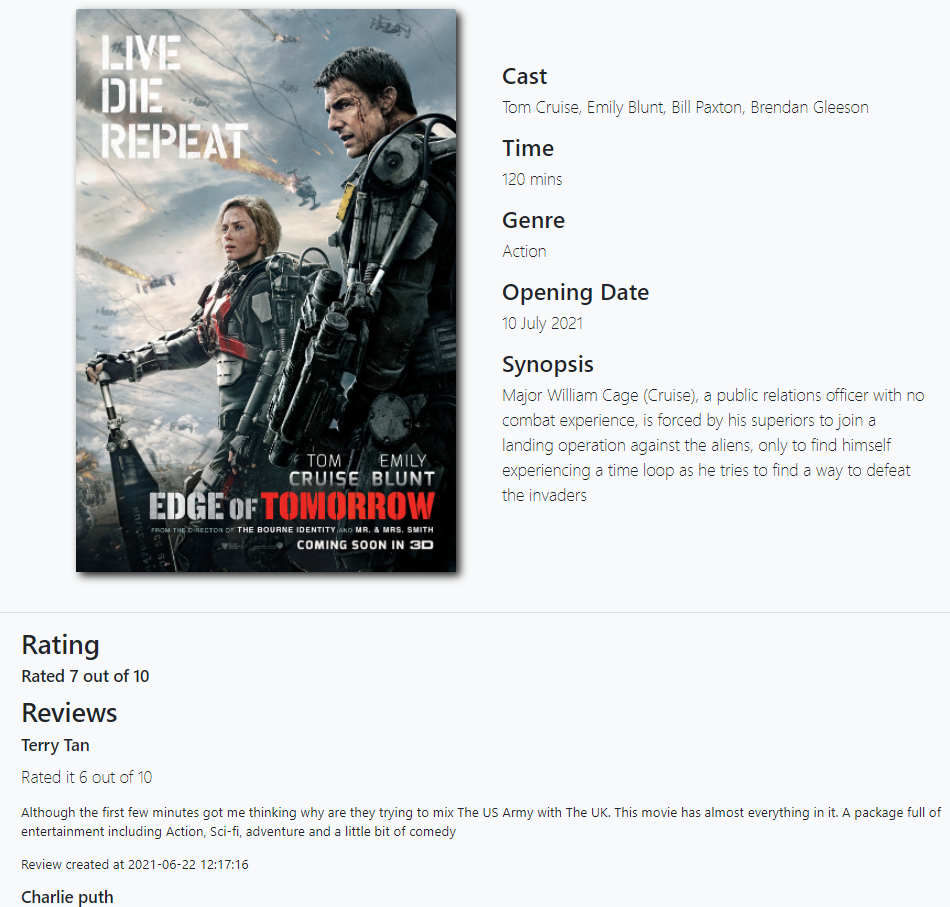
When a user clicks on the read more button on a movie listing on the home page, they will be taken to another page where they can view that specific movie’s details. The front-end will first load the movieDetails.html page, which will then call the axios.get(moviesByID) function to load the movie’s details from the back end



Aside from that, we have also created an API to get all the reviews for that specific movie on the same page. After the axios.get(moviebyID) function has finished retrieving the data, the next axios function, axios.get(movieReviews) will get the corresponding reviews for the movie displayed by calling that function while passing the movieID to the back end.

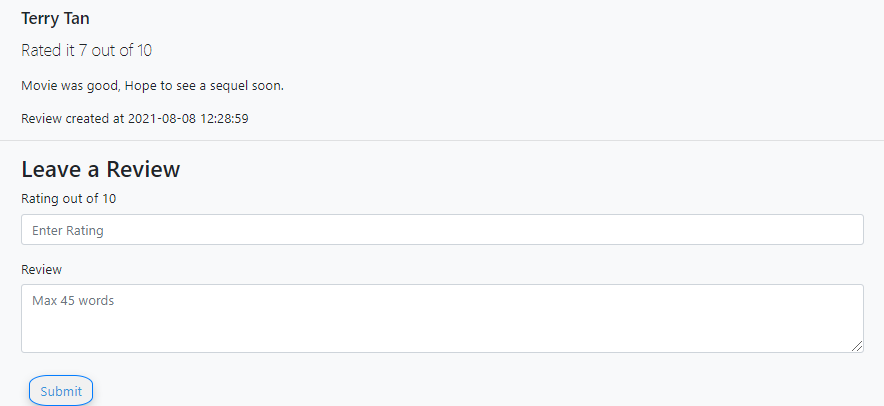


This is what the front-end will look like



## POST movie review

While on the movieDetails.html page, the user also has the ability to leave a review for that movie.



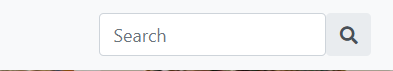
The user can fill up the form with their rating and review, then submit it.



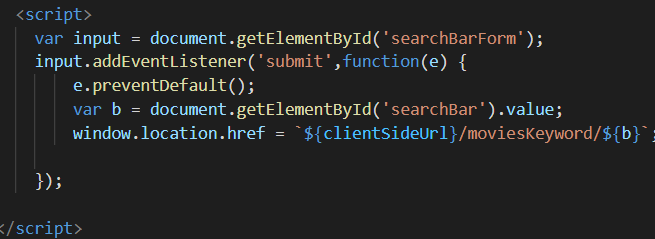
The data from the form will be passed to the front-end server where it will be POST to the back-end using axios.

## GET movie by KEYWORD, search bar

At the top of the home page, there is a search bar users can use to search a movie by keyword



When the user inputs a word and presses enter, the data will be passed to a script in home.html, which will load the searchedByKeyword.html page.

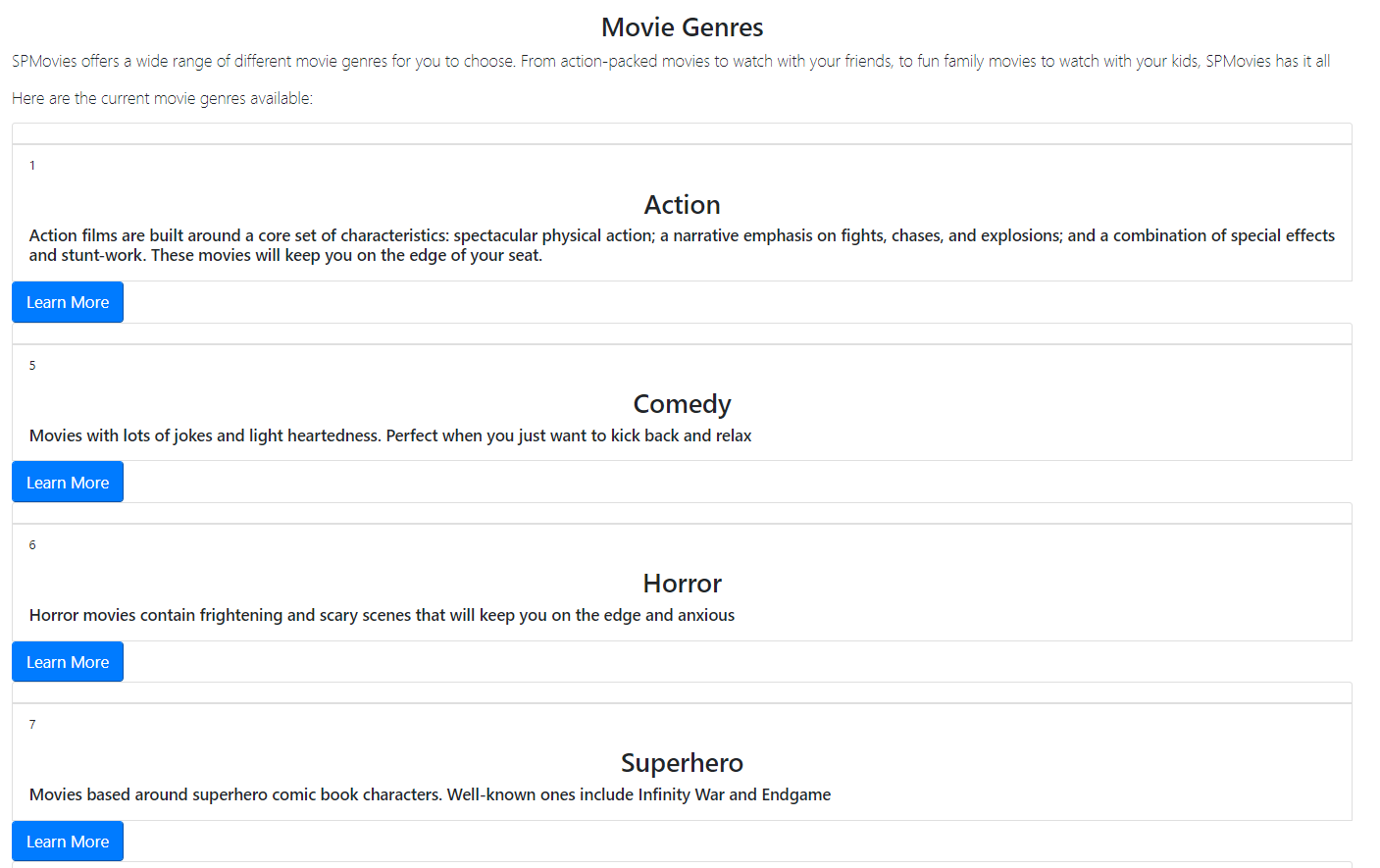


When the user reaches the searchedByKeyword.html page, the keyword will be passed from the url to the axios.get function where it will be sent to the back-end to search for a movie that corresponds to the searched keyword.



## GET genres

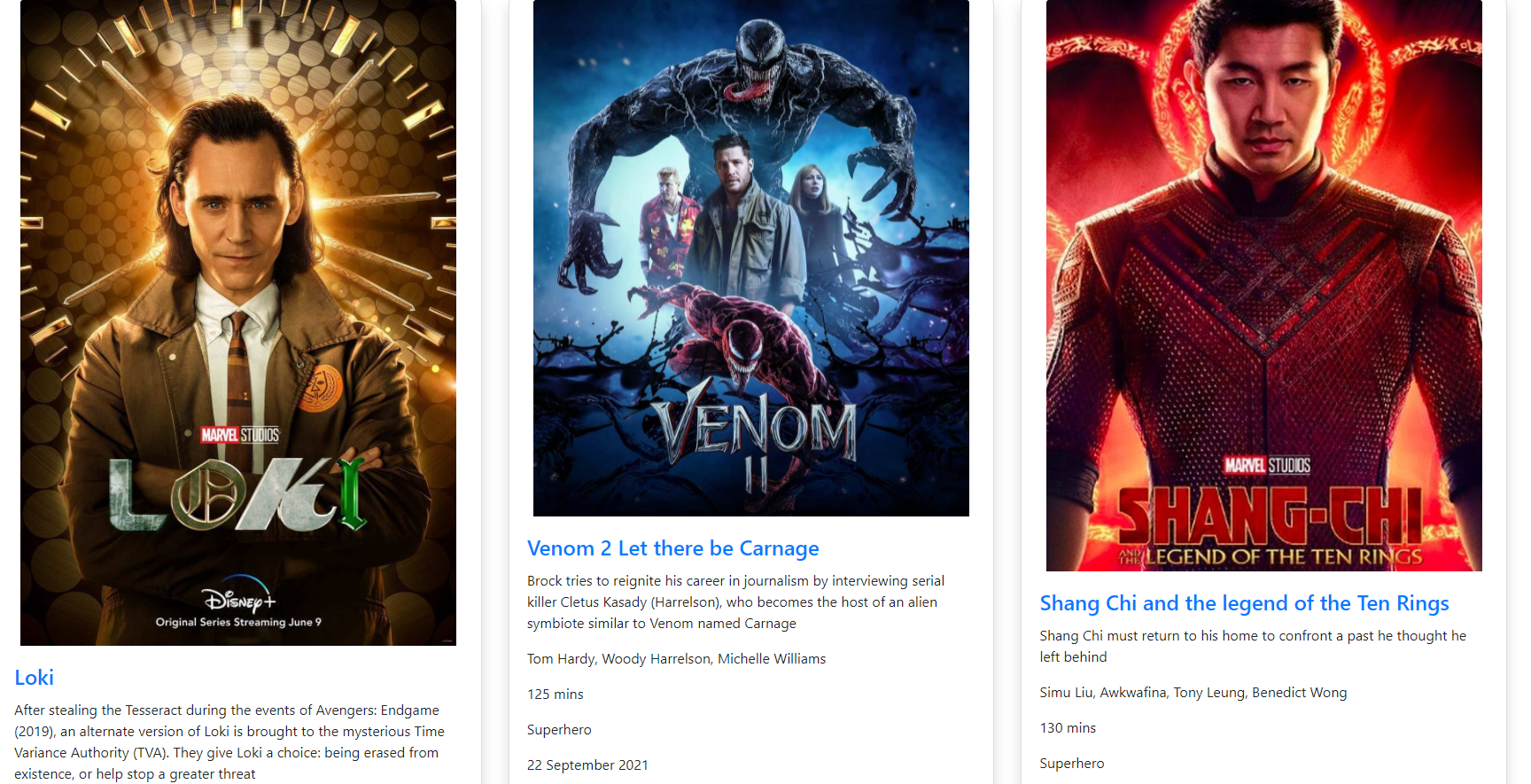
We have also created a genres.html page that will show all genres in the database. When the page is loaded, the axios.get(genresall) function will load, getting all the genres from the database.



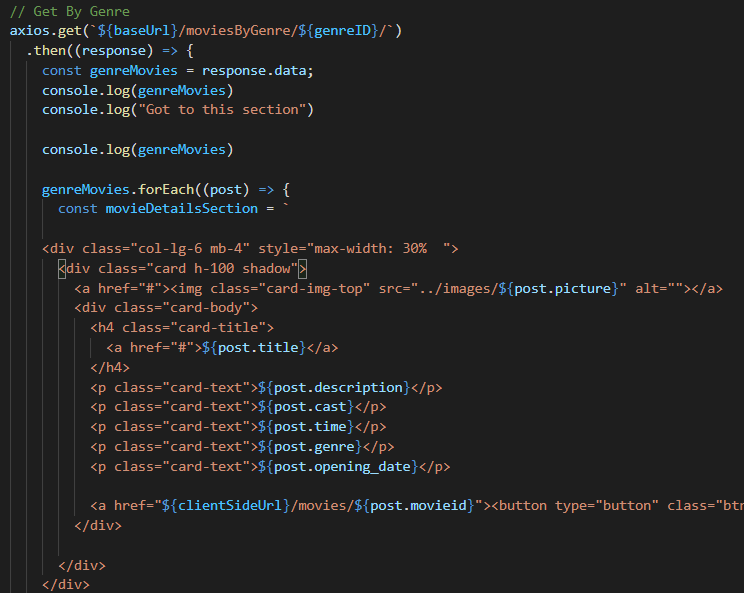


## GET movies By GenreID

If users click on the learn more button on a genre listing, they will be taken to the searchedMovies.html page which will list all the movies that belong to that genre.

For example, if the user clicks on the superhero genre, all movies belonging to the superhero genre will be shown.

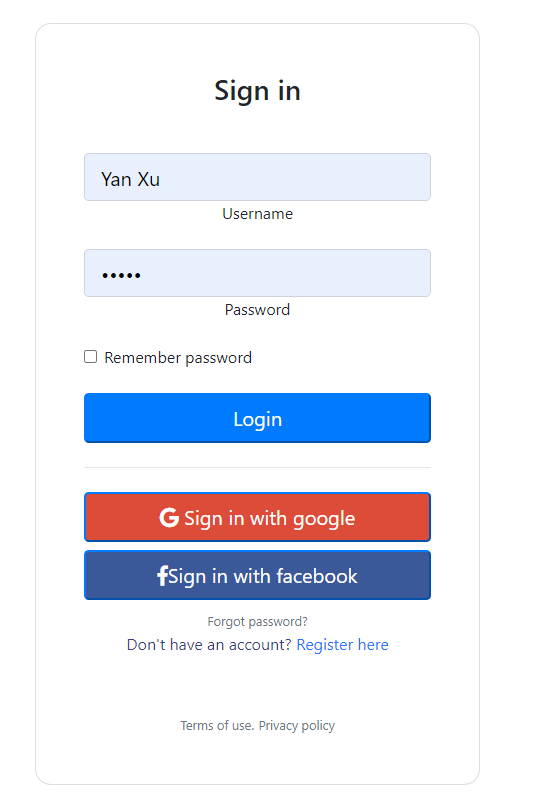
The genreID will be passed into the axios.get(moviesByGenre) function, which will then pass the data to the backend to search by genreID



## POST Login api

We have also created a Login api which supports the creation of user accounts for the website. Users can log in to leave reviews for a movie.

The user can enter their username and password into this sign in form



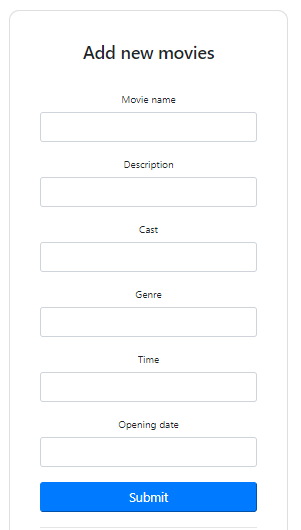
The sent data will be passed to the front-end where it will be sent using the axios.post(login) function to the back-end



## Admin Page

We also have an admin page where admins can create new movies, new genres, edit a movie listing or delete a movie.

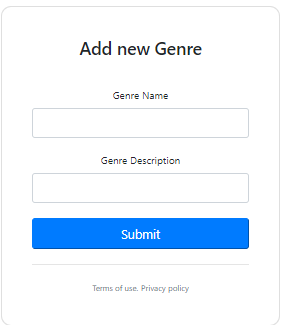
**Add movie**



The data will be passed to the axios.post(movies) function to the back-end to store the movie in the database



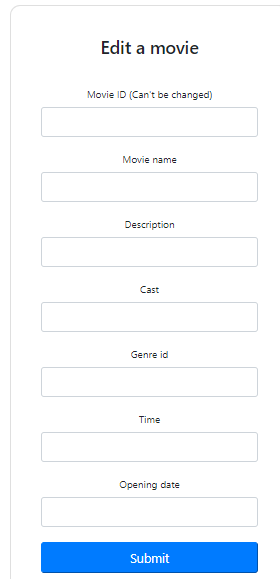
**Add genre**

****

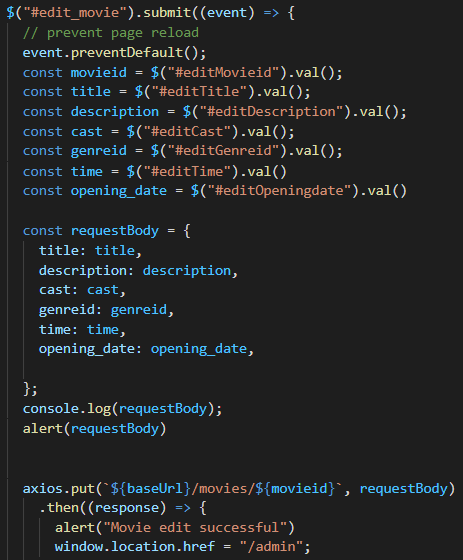
The data will be passed to the backend using the axios.post(genres) function

****

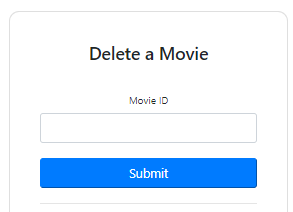
**Edit movie**

****

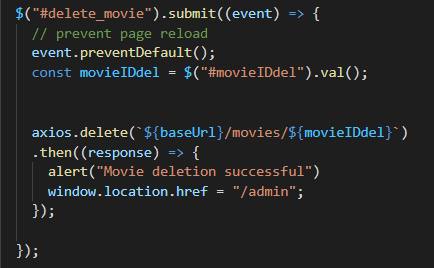
The data is passed to the backend using the axios.put(movieid) method

****

**Delete movie**

****

The data is passed to the backend using the axios.delete(movieid) method

****